Chapter 1: Introduction to UNIX at Fermilab

This chapter is intended to introduce you to the UNIX environment, the UNIX file systems and the computing security in use at Fermilab.

1.1 Computer Security at Fermilab

1.1.1 Strong Authentication and Kerberos v5

In order to protect against unauthorized access to Fermilab computers, the Computing Division has implemented the Kerberos Network Authentication Service V5, developed at MIT, to provide what is known as *strong authentication* over the network.

"Authentication" refers to verifying the identities of networked users, clients and servers. "Strong" authentication is a means of verifying these identities without transmitting passwords over the network, and without requiring that the network itself be protected.

Kerberos v5 is the strong authentication program that Fermilab computers are required to run. Kerberos authenticates users by way of exchanging electronic tickets between clients and services. It cleverly encrypts and de-encrypts these tickets before and after transmitting them. A machine on which Kerberos v5 has been installed and which enforces the Kerberos authentication is referred to as a *strengthened* or *Kerberized* machine.

The "heart" of a Kerberos system is the Key Distribution Center (KDC), which maintains a database of member computers and users, and grants authentication requests. The set of member computers make up what's called a "strengthened realm". At Fermilab, the strengthened realm for UNIX machines is called FNAL.GOV.

All UNIX machines at Fermilab are required to be configured such that they are members of the FNAL.GOV realm. Off-site machines used for Fermilab-related work may also be configured as such.

Once you have authenticated to the FNAL.GOV realm on your desktop, you can freely access over the network any computer in this realm on which you have an account, without retyping your (FNAL.GOV) Kerberos password!

The Fermilab Strong Authentication implementation is described in the *Strong Authentication at Fermilab* website and manual, online at (`http://www.fnal.gov/docs/strongauth/`). Try out the "Getting Started" link.

1.1.2 Fermilab Policy and Your Responsibilities

As a computer user at Fermilab, you are required to read the **Fermilab Policy on Computing**; see `http://www.fnal.gov/cd/main/cpolicy.html`.

The Kerberos authentication system exercises tight control over who uses the lab's computers and network, but as with any security system, it requires those with legitimate access to "lock the doors behind themselves" and "keep the key in a safe place". In our computing environment this translates into a set of responsibilities for all Fermilab computer users. These responsibilities are listed in the Computing portion of the **New Employee Orientation** under "Fermilab Policy on Computing" at `http://computing.fnal.gov/orientation/policy.html`. This page is also accessible via a link from the "Getting Started" portion of the Strong Authentication manual, `http://www.fnal.gov/docs/strongauth/getstart.html`.

1.1.3 Conditions of Use for Federal Computers

The DOE requires that Fermilab computers post a notice to users regarding the conditions of use. The message typically prints to screen after a successful login. Make sure you understand and accept the conditions of use before you proceed.

1.2 A Few Words about UNIX

Because UNIX was originally designed by programmers to support their own projects, one of its strongest points is that it provides an excellent software development environment. UNIX has a large set of powerful utility programs and tools that allow users to easily build systems and applications. It also has several command interpreters, called *shells* that can be used as high-level programming languages.

1.2.1 Variations of the UNIX OS

There are several *flavors* of the UNIX operating system, corresponding roughly to individual vendors' hardware platforms, e.g., SunOS from Sun Microsystems, IRIX from SGI, and so on. The Linux flavor breaks the mold, as it can be installed on a number of platforms. The flavors are in large part quite similar, although various commands, files and other features may work somewhat differently from one to another. In this manual, the commands shown are valid for all supported flavors, except when noted otherwise.

1.2.2 UNIX OS Components

The UNIX operating system has four basic components:

- The *kernel* constitutes the nucleus of the operating system and coordinates the internals such as allocating system resources.
- The *file system*, which is part of the kernel, controls the storage and access of data.
- *Commands* are programs that the computer executes upon command.
- Programs called *shells* serve as command interpreters. They act as links between user and computer, interpreting and executing commands. They are also high-level interpretive programming languages. There are two commonly-used families of shells: Bourne and Berkeley/C.

1.3 The Fermi UNIX Environment (FUE) and Product Support

The Fermilab Computing Division (CD) supports several UNIX operating system flavors; see the document *Certified UNIX Operating Systems* at `http://www.fnal.gov/docs/Recommendations/dr0010.html` for a current listing. Across this variety of UNIX flavors, the CD makes available a dependable, easy-to-use and reasonably uniform computing environment known as the *Fermi UNIX Environment*, or *FUE* for short. One of the main goals of FUE is to provide as much as possible the same environment on the different UNIX platforms that is dependable and easy to use. The other primary goal is to provide a standard product support methodology and toolkit, to which end the CD has developed the UNIX Product Support and UNIX Product Distribution (UPS/UPD) suite of tools. There are two levels of FUE: CoreFUE and FullFUE.

The UPS/UPD tools are documented in the *UPS, UPD and UPP v4 Complete Guide and Reference Manual*, found online at `http://www.fnal.gov/docs/products/ups/`.

1.3.1 CoreFUE

The combination of the products UNIX Product Support (UPS) and UNIX Product Distribution (UPD)[1] is called CoreFUE. CoreFUE provides the Fermilab UNIX software support environment, leaving the rest of the user environment essentially untouched. It can be installed independently, without the other portions of FUE. CoreFUE is well-suited to off-site systems that interact with Fermilab computers and to which users download Fermilab products.

1.3.2 FullFUE

For on-site systems, we recommend the installation of FullFUE, which consists of CoreFUE plus the following products:

systools	Local system administration utilities. Systools includes a suite of smaller products. The features you get with systools include: the FUE login scripts to standardize your environment, the "**`cmd`**" command which is used to allow particular users or groups to run commands with superuser or other privilege, and the bash and tcsh shells.
futil	Fermilab-specific utilities including flpr (see section 11.2 *The Print Commands*, psutils, telephone, stock, and gtools

To get information about any of these individual products, go to the Computing Division home page (`http://www.fnal.gov/cd/`) or documentation search page (`http://cddocs.fnal.gov/cfdocs/productsDB/docs.html`), and look them up.

1.3.3 The Login Files for FullFUE

If you're running FullFUE, a series of scripts is run when you log in that define the functionality of your terminal and set up your environment. These script files are located in your home directory. You can peruse the default file listings in `/usr/local/etc/`. The file names differ depending on the shell you

1. Perl is also part of CoreFUE, as it is a required component of UPS/UPD.

use; shells are described in Chapter 5: *Shells*. The shell runs (usually) two files. Shells of the C shell family, csh and tcsh, run the two files `.cshrc` and `.login`. There is more variety among the Bourne family shells:

sh	runs `.profile`
ksh	runs `.profile` and `.bashrc`
bash	runs `.profile` and (sometimes) `.kshrc` in place of `.shrc`

Sometimes (more common recently) vendors have linked sh to ksh, effectively replacing sh with ksh.

If your account was created under FullFUE, these files are Fermilab-specific. How can you tell? For C shell, look for the following text in your `.login` (the path to `setups.[c]sh` depends on where UPS is installed):

```
if ( -f "/afs/fnal.gov/ups/etc/setups.csh" ) then
    source "/afs/fnal.gov/ups/etc/setups.csh"
    if ( { ups exist login } ) then
        setup login
    endif
endif
```

For Bourne shell, look for this in your `.profile`:

```
if [  -f "/afs/fnal.gov/ups/etc/setups.sh" ]
then
    . "/afs/fnal.gov/ups/etc/setups.sh"
    if ups exist login
    then
        setup login
    fi
fi
```

1.3.4 The Command Prompt

The default UNIX prompt usually indicates your default shell, where typically, **`%`** indicates the C shell (**csh**) and **`$`** indicates the Bourne shell (**sh**). However the FullFUE login files set the machine prompt to the machine name, e.g., **`<fsui02>`**.

You can change your default prompt by altering your start-up files (note the intentional space between the new prompt name and the closing angle bracket):

- For C shell, include in your `.login` file:

  ```
  set prompt='<newprompt >'
  ```

- For Bourne shell, you need to set the value of the keyword shell variable PS1. This is just a variable that is declared and initialized by the shell at start-up. Include in your `.profile` file:

  ```
  export PS1; PS1="<newprompt >"
  ```

1.4 File Systems: Standard UNIX and AFS

1.4.1 Standard UNIX File System

The standard UNIX file system is a hierarchy of directories descending from what is known as the *root directory.* UNIX allows parts of the directory hierarchy to reside on separate storage devices or in separate disk partitions. These separately stored parts are called *file systems*. They are accommodated in the main hierarchy by means of *mount points*. A mount point is a directory in a file system that corresponds to the `root` directory of some other file system. A machine's or cluster's primary file system is the one starting at the true `root`. The standard UNIX file system is described in Chapter 7: *The UNIX File System*.

1.4.2 AFS File System

If the UNIX machine that you work on is part of an integrated system of UNIX machines at Fermilab, for example a LAN (local area network), it is likely that the AFS distributed file serving system has been installed on it.

AFS is a shared file system. "AFS space" is a UNIX directory/file area starting at `/afs` that can be shared between computers. This is handy when large numbers of people need to access files in an area. Fermilab has a "cell" in AFS space, which is simply the area under the AFS root directory belonging to Fermilab. Fermilab's cell is `/afs/fnal.gov/`. Any directory under this cell, e.g., `/afs/fnal.gov/x/y/z/`, has exactly the same contents when viewed/manipulated on one computer as on another, provided both computers implement AFS with the Fermilab cell. Users are required to authenticate to AFS.

See Chapter 8: *The AFS File System* for a discussion of AFS. Section 8.2 *How to Determine if AFS is Installed on your System* describes how to determine if AFS is installed on your machine.